CS10720 Problems and Solutions

Thomas Jansen

Today: Representing Rational Numbers

February 4^{th}

Reminder

Don't forget portfolio submission deadline tomorrow, 7pm
containing

- summary of Monday's lecture (representing integers)
- answers to both questions in one of the three problems from the worksheet about representing integers from the practicals
- summary of today's lecture (representing rational numbers)

Ideally everything in only one blog entry
but clearly marking what is what

Reminder portfolio contributes 40% to module marks
meaning you will fail the module if you don't do it

Plans for Today

Rational Numbers

Remember we have binary representations for natural numbers $x \in \mathbb{N}$ and for integers $x \in \mathbb{Z}$ (signed magnitude; one's complement; two's complement; excess)

What about real numbers $x \in \mathbb{R}$?

Observation in general, not feasible
since we're dealing with finite computers/representations

What about rational numbers $x \in \mathbb{Q}$?

Remark two different possibilities
fixed point representations
i. e., fixed number of digits after decimal point
+simple
−useful only in special circumstances (e. g., money: £2.98)
floating point representations

Towards Floating Point Representations

For starters decimal representations

Observation almost every number $x \in \mathbb{Q}$ has unique representation as
$x = (-1)^s \cdot m \cdot 10^e$
with sign $s \in \{0, 1\}$
exponent $e \in \mathbb{Z}$
mantissa $m \in \mathbb{Q}$ and $1 \leq m < 10$
is called normalised floating point representation

Examples and Observations

- $12.34 = (-1)^0 \cdot 1.234 \cdot 10^1$
- $-383.902 = (-1)^1 \cdot 3.83902 \cdot 10^2$
- $-0.023 = (-1)^1 \cdot 2.3 \cdot 10^{-2}$
- $9.1 = (-1)^0 \cdot 9.1 \cdot 10^0$
- 0 cannot be represented this way

Binary Floating Point Representations

Observation $x = (-1)^s \cdot m \cdot 10^e$ is decimal representation
we prefer binary representation
$x = (-1)^s \cdot m \cdot 2^e$ with $1 \leq m < 2$
(m and e in binary representation)

Example $(1011.101)_2 = (-1)^0 \cdot (1.011101)_2 \cdot 2^3$

Observation cannot represent $(0.2)_{10}$ like this

Proof Assume $0.2 = \frac{1}{5} = \sum_{i=-l}^{-3} d_i \cdot 2^i$ with $d_i \in \{0, 1\}$

$$\text{thus } \frac{1}{5} = \sum_{i=3}^{l} d_{-i} \cdot 2^{-i} = 2^{-l} \sum_{i=3}^{l} d_{-i} \cdot 2^{l-i}$$

$$= 2^{-l} \sum_{i=0}^{l-3} d_{-l+i} \cdot 2^i = \frac{a}{2^l} \text{ with } a, l \in \mathbb{N}$$

thus $5a = 2^l$ $\lightning$ $\square$

About the Example

Remember $(0.2)_{10}$ cannot be represented in binary with any finite number of digits

Is this a specific problem with the binary representation?

Observation for 0.2 it is, in general it is not

Example Consider $\frac{1}{3}$.
$\frac{1}{3} = (0.1)_3$
but $\frac{1}{3}$ cannot be represented in decimal
with any finite number of digits

Conclusion many rational numbers have no finite floating point rep.
and that holds for any base
$\rightsquigarrow$ we can proceed with base 2

IEEE-754

Institute of Electrical & Electronical Engineers

Basis $x = (-1)^s \cdot m \cdot 2^e$
with sign $s \in \{0, 1\}$
exponent $e \in \mathbb{Z}$
mantissa $m \in \mathbb{Q}$ and $1 \leq m < 2$

Definitions

- mantissa m represented in standard binary encoding
- do not store leading 1 in m (called implicit 1)
- exponent e in excess representation with $k = 2^{l_e - 1} - 1$

Caution artificially restrict exponent further (by 1 'on each side')

$$e_{\min} = -k + 1 = -\left(2^{l_e - 1} - 1\right) + 1$$

$$e_{\max} = 2^{l_e} - 1 - k - 1 = 2^{l_e - 1} - 1$$

(to make room for encoding 'special' numbers)

What is l_e?

IEEE-754

Remember $x = (-1)^s \cdot m \cdot 2^e$
with sign $s \in \{0, 1\}$
exponent $e \in \mathbb{Z}$ in excess representation with $k = 2^{l_e - 1} - 1$
e further restricted between $e_{\min} = -k + 1 = -\left(2^{l_e - 1} - 1\right)$
and $e_{\max} = 2^{l_e} - 1 - k - 1 = 2^{l_e - 1} - 1$ to make room
to encode 'special' numbers
mantissa $m \in \mathbb{Q}$ and $1 \leq m < 2$, rep. in binary encoding

Length of Representation

name	total length	sign	exponent	mantissa
single precision	32	1	8	23
double precision	64	1	11	52
quadruple precision	128	1	15	112

IEEE 754: Special Numbers

'number'	sign	exponent	mantissa
$+\infty$	0	$e_{\mathsf{max}} + 1$	0
$-\infty$	1	$e_{\mathsf{max}} + 1$	0
NaN	s	$e_{\mathsf{max}} + 1$	$\neq 0$
0	0	$e_{\mathsf{min}} - 1$	0
-0	1	$e_{\mathsf{min}} - 1$	0
$(-1)^s \cdot \left(\sum_{i=1}^{l_m} m_i \cdot 2^{-i}\right) \cdot 2^{e_{\mathsf{min}}}$	s	$e_{\mathsf{min}} - 1$	$m \neq 0$

NaN not a number

last row denormalised representation

Why denormalised numbers?

About Denormalised Numbers

Obvious can represent even smaller numbers

Example `If` $x \neq y$ `Then` $z := 1/(x-y)$

$x = 0\,0000\,0001\,000\,0000\,0000\,0000\,0000\,0001$ $2^{-126} + 2^{-149}$

$y = 0\,0000\,0001\,000\,0000\,0000\,0000\,0000\,0000$ 2^{-126}

Obvious $x \neq y$

But without denormalised representation $x - y = 0$ rounded

denormalised $x - y = 2^{-149}$:

$0\,0000\,0000\,000\,0000\,0000\,0000\,0000\,0001$

Always better `If` $x - y \neq 0$ `Then` $z := 1/(x-y)$

IEEE 754: An Example

$l = 32$, $l_s = 1$, $l_e = 8$, $l_m = 23$
$k = 2^7 - 1 = 127$, $e_{\mathsf{min}} = -k + 1 = -126$,
$e_{\mathsf{max}} = 2^8 - k - 2 = 127$

We want to encode -3.

negative, thus sign 1

binary representation $3 = 2 + 1 = 2^1 + 2^0 = (2^0 + 2^{-1}) \cdot 2^1$

exponent 1 excess rep. $\rightsquigarrow$ represent $1 + k = 128$

$128 = (1000\,0000)_2$

mantiassa 1.1, implicit 1 not represented, thus $100\,0000\,0000 \cdots$

Result $\underbrace{1}_{\text{sign}}\underbrace{1000\,0000}_{\text{exponent}}\underbrace{100\,0000\,0000\,0000\,0000\,0000}_{\text{mantissa}}$

IEEE 754: Another Example

$l = 32$, $l_s = 1$, $l_e = 8$, $l_m = 23$
$k = 2^7 - 1 = 127$, $e_{\text{min}} = -k + 1 = -126$,
$e_{\text{max}} = 2^8 - k - 2 = 127$

We want to encode 0.0546875.

positive, thus sign 0

binary representation $0.0546875 = \frac{1}{32} + \frac{1}{64} + \frac{1}{128} = 2^{-5} + 2^{-6} + 2^{-7} = \left(2^0 + 2^{-1} + 2^{-2}\right) \cdot 2^{-5}$

exponent -5 excess rep. $\leadsto$ represent $-5 + k = 122$

$122 = (0111\,1010)_2$

mantiassa 1.11, implicit 1 not represented, thus $110\,0000\,0000 \cdots$

Result $\underbrace{0}_{\text{sign}}\,\underbrace{0111\,1010}_{\text{exponent}}\,\underbrace{110\,0000\,0000\,0000\,0000\,0000}_{\text{mantissa}}$

IEEE 754: Final Example

$l = 32$, $l_s = 1$, $l_e = 8$, $l_m = 23$
$k = 2^7 - 1 = 127$, $e_{\mathsf{min}} = -k + 1 = -126$,
$e_{\mathsf{max}} = 2^8 - k - 2 = 127$

We have $0\,1000\,0101\,010\,1001\,0000\,0000\,0000\,0000$.

$$\underbrace{0}_{\text{sign}}\ \underbrace{1000\,0101}_{\text{exponent}}\ \underbrace{010\,1001\,0000\,0000\,0000\,0000}_{\text{mantissa}}$$

sign 0, thus positive

exponent $(1000\,0101)_2 = 133$, represents $133 - k = 133 - 127 = 6$

mantissa including implicit 1 is 1.0101001
$(1.0101001)_2 = 1 + \frac{1}{4} + \frac{1}{16} + \frac{1}{128}$

thus $\left(2^0 + 2^{-2} + 2^{-4} + 2^{-7}\right) \cdot 2^6 = 2^6 + 2^4 + 2^2 + 2^{-1} = 84.5$

http://onlineted.com

Summary & Take Home Message

Things to remember

- fixed point representations
- floating point representations
- floating point representations with different bases
- IEEE 754 floating point representation
  - definition
  - representation lengths
  - special numbers
  - denormalised representation
  - examples

Take Home Message

- IEEE-754 representation for rational numbers combines different representations.
- Standards are important.

Lecture feedback `http://onlineted.com`